

Ville-Valtteri Pennala

Veneen sähkömoottoriohjausjärjestelmän kehittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Kone- ja tuotantotekniikka

Insinöörityö

29.4.2015

Tekijä Otsikko	Ville-Valtteri Pennala Veneen sähkömoottoriohjausjärjestelmän kehittäminen
Sivumäärä Aika	19 sivua + 2 liitettä 29.4.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto	Koneautomaatio
Ohjaajat	Projektin johtaja Mikko Säaskilahti Lehtori Heikki Paavilainen
<p>Insinööriytyössä suunniteltiin ja kehiteltiin veneenohjausjärjestelmää kaksimoottorijärjestelmässä Black Cat Boats Oy:lle. Järjestelmää on aikaisemmin kehitelty Black Cat Boatsin toimesta ja Metropolian Ammattikorkeakoululla TKI-projekteina. Järjestelmäkoonpano on suunniteltu aikaisemmin em. projektien yhteydessä. Järjestelmään on myös kirjoitettu lähdekoodi aikaisemmin.</p> <p>Järjestelmään suunniteltiin uusi kokoonpano, joka jäi kuitenkin toteuttamatta, sillä tarkoitus oli päästä testaamaan järjestelmää käytännössä ja se onnistuu nykyiselläkin kokoonpanolla. Nykyisestä kokoonpanosta vaihdettiin moottoreiden asennontunnistukseen käytettävä menetelmä. Erilaiset ohjaukseen tarvittavat algoritmit suunniteltiin uudelleen matemaattisesti vektoriperiaatteella. Ohjauksen toteuttavan mikrokontrollerin lähdekoodi uusittiin.</p> <p>Tämän työn kirjoitushetkellä järjestelmää ei ole vielä testattu veneessä, sillä järjestelmä ei vielä ole sopiva veteen rakenteellisista syistä johtuen. Aikataulu on myös vaikuttanut siihen, ettei käytännön testejä ole ehditty suorittamaan. Järjestelmää on testattu testipenkissä, ja siinä se toimii oletetusti. Järjestelmän toiminta perustuu matemaattiseen teoriaan, mutta sitä ei ole kokeiltu käytännössä. Insinööriytyön ulkopuolelle jäi aikataulullisista syistä myös graafinen käyttöliittymä (GUI), josta olisi suuri etu järjestelmälle.</p>	
Avainsanat	Vene, sähkömoottori, ohjausjärjestelmä

Author Title	Ville-Valtteri Pennala Twin Motor Steering System for Black Cat Boats
Number of Pages Date	19 pages + 2 appendices 29 April 2015
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Specialisation option	Machine Automation
Instructors	Mikko Sääskilahti, Project manager Heikki Paavilainen, Lecturer
<p>In this Bachelor's thesis the objective was to design and develop an electric motor boat steering system for a twin motor layout. The assignment was carried out for Black Cat Boats. Most of the hardware in this system has already been designed and made by Black Cat Boats as the result of various research and development projects carried out in Metropolia University of Applied Sciences. The source code for the control system was made in a previous research project.</p> <p>A new system layout was designed for the system, but it was not carried out since the main objective was to test the system and this could be achieved with the existing one. The element for motor position feedback was replaced by another type of component. Various algorithms for steering the vessel were redesigned for mathematical equations relying on the vector based system. The source code for the control system was updated.</p> <p>By the time of this thesis the control system is yet to be tested in the water, because of a few mechanical flaws. Also the timeline of this project has contributed to the fact, that this system has not been tested properly yet. The system was tested on a test bench and it performed as expected. The functionality of this system relies on the mathematical theory. A graphical user interface desired for this system and it would have contributed a lot to it, however due to the busy schedule it was not examined in this thesis.</p>	
Keywords	Boat, electric motor, steering system

Sisällys

Lyhenteet

1 Johdanto.....	1
2 Järjestelmän kuvaus.....	1
3 Ohjausjärjestelmän määrittely.....	2
3.1 Valmiustila.....	3
3.2 Telakointi-ajotapa.....	4
3.3 Ensimmäinen ajotapa.....	7
3.4 Toinen ajotapa.....	8
4 Toimilaitteet ja kokoonpano.....	9
5 Ohjelma sulautetussa ympäristössä.....	11
5.1 Arduino Uno-mikrokontrolleri projektissa.....	12
5.2 Arduino IDE.....	13
6 Jatkokehitysmahdollisuudet.....	14
6.1 Näyttö ja GUI.....	15
6.2 Taivasankkuri.....	15
6.3 GPS ja modulaarinen ohjelma.....	15
6.4 Mikrokontrollerit ja lähdekoodin kääntäminen.....	16
6.5 Järjestelmäkokoonpanon muutokset.....	16
7 Prototyypin testaus.....	17
8 Yhteenveto.....	19
Liite 1. Moottoreiden momentin jakautuminen Black Cat Boats katamaraanissa	
Liite 2. Sähkökaavio	

Lyhenteet

CAN	Controlled Area Network. Automaatioväylätekniikka.
EEPROM	Electrically Erasable Programmable Read-Only. Mikroprosessorin pysyvä muisti.
GPS	Global Positioning System. Satelliittipaikannusjärjestelmä.
GUI	Graphical User Interface. Graafinen käyttöliittymä.
IDE	Integrated Development Environment. Kehitysympäristö, jossa on kääntäjä (compiler) samassa.
I ² C ja SPI	Inter-Integrated Circuit ja Serial Peripheral Interface. Väylätekniikoita tiedonsiirtoon.
NMEA	National Marine Electronics Association. Standardiliitto, joka erikoistuu merenkulun kommunikaatioon.
PWM	Pulse Width Modulation. Tapa jäljitellä analogista signaalia digitaalitekniikassa.

1 Johdanto

Insinööriyössä on tarkoituksena kehittää Black Cat Boatsille veneen ohjausjärjestelmä kahdelle moottorille. Ohjausjärjestelmän tulisi mahdollistaa veneen ohjaus käyttäjäystävällisesti tilanteesta riippuen. Usein veneen telakointi on haastavaa kokemattomille kuljettajille. Tämän järjestelmän on tarkoitus helpottaa veneen telakointia. Ohjausjärjestelmä sisältää myös muihin veneilytilanteisiin tarkoitettuja ajotapoja, mutta telakointi-ajotapa on näistä tärkein ominaisuus. Ohjaus tapahtuu sauvaohjaimella.

Työtä on tehty aikaisemmin TKI-projekteina. Aikaisemmissa projekteissa on valittu järjestelmän toimilaitteet ja kokoonpano. Tässä työssä vaihdettiin alkuperäisessä kokoonpanossa olleet asennon indikoimiseen tarkoitetut induktiiviset anturit absoluuttienkooderiin. Työn pääpaino on järjestelmää käyttävän ohjelman suunnittelussa ja toteuttamisessa, ottaen huomioon ohjausjärjestelmän soveltuvuuden erilaisiin venetyyppeihin. Projekti toteutetaan prototyypikokoonpanolla Black Cat Boats-katamaraaniin, joten järjestelmän kokoonpano saattaa muuttua myöhemmin mahdollisen tuotteistuksen yhteydessä.

Black Cat Boats Oy on Rovaniemellä toimiva yritys, joka valmistaa alumiinirunkoisia huviveneitä. Yritys on perustettu 2011, ja sillä on tällä hetkellä kaksi venettä mallistossaan. Tätä ohjausjärjestelmää kehitellään pienemmälle katamaraanirunkoiselle veneelle.

2 Järjestelmän kuvaus

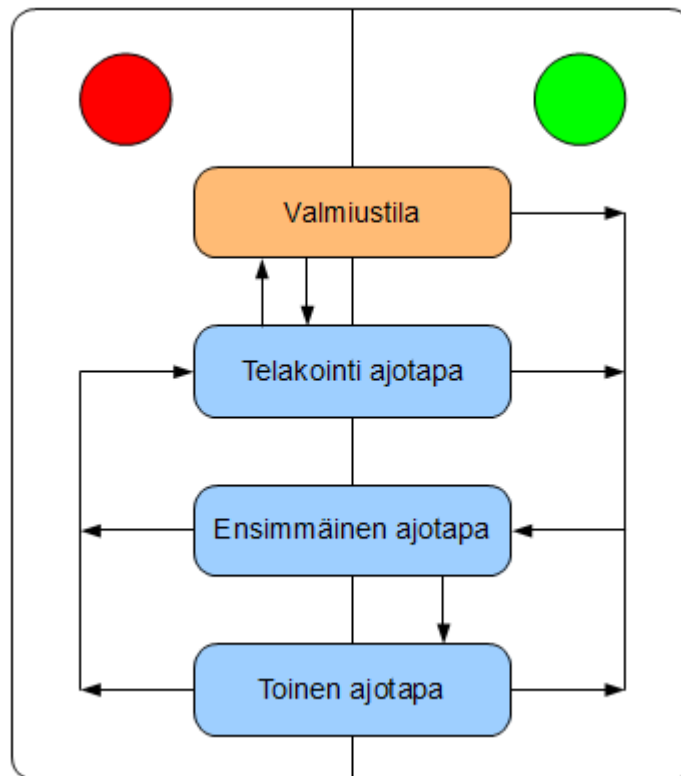
Järjestelmässä on kaksi sähköistä perämoottoria, joita voidaan kääntää itsenäisesti. Järjestelmä toimii 24 V:n tasavirtajännitteellä. Ohjaus tapahtuu ohjaussauvalla ja ajotapa valitaan painonapeilla. Ohjaussignaalit menevät mikrokontrollerille, joka muuntaa ne teho- ja suuntaohjeiksi ajotavasta riippuen. Nämä ohjeet lähetetään moottorinohjausyksiköille, jotka käyttävät moottoreita. Suuntaohjaus on asematakaisinkytketty ja teho säädetään proportionaalisesti ohjaussauvan asennosta riippuen.

Järjestelmä on suunniteltu pienille ja keskisuurille veneille, joita on mahdollista ajaa sähkömoottorin avulla. Black Cat Boatsin katamaraanissa moottorit saadaan nostettua ylös vedestä, jolloin voidaan käyttää veneen perän keskellä sijaitsevaa polttoainekäyttöistä perämoottoria. Sähkömoottorit on tarkoitettu veneen telakointiin, sekä hitaaseen liikkumiseen, kuten uisteluun.

3 Ohjausjärjestelmän määrittely

Ohjausjärjestelmä koostuu kolmesta toisistaan poikkeavasta ajotavasta ja valmiustilasta, jossa ei tapahdu ohjausliikkeitä järjestelmän ollessa kuitenkin aktiivisena.

Ohjaustapa valitaan kahdella painonapilla (punainen ja vihreä). Päällä olevaa ajotapaa indikoidaan kahdella vastaavan värisellä LED merkkivalolla. Valmiustilassa molemmat merkkivalot ovat sammuneina. Telakointi-ajotavan ollessa kytkettynä punainen merkkivalo palaa. Ensimmäistä ajotapaa indikoi koko ajan palava vihreä merkkivalo, ja toisessa ajotavassa vihreä merkkivalo vilkkuu (kuva 1).



Kuva 1: Kaavio esittää ajotavan valinnan. Vasemmalla punainen painonappi, oikealla vihreä painonappi ja keskellä ajotapa.

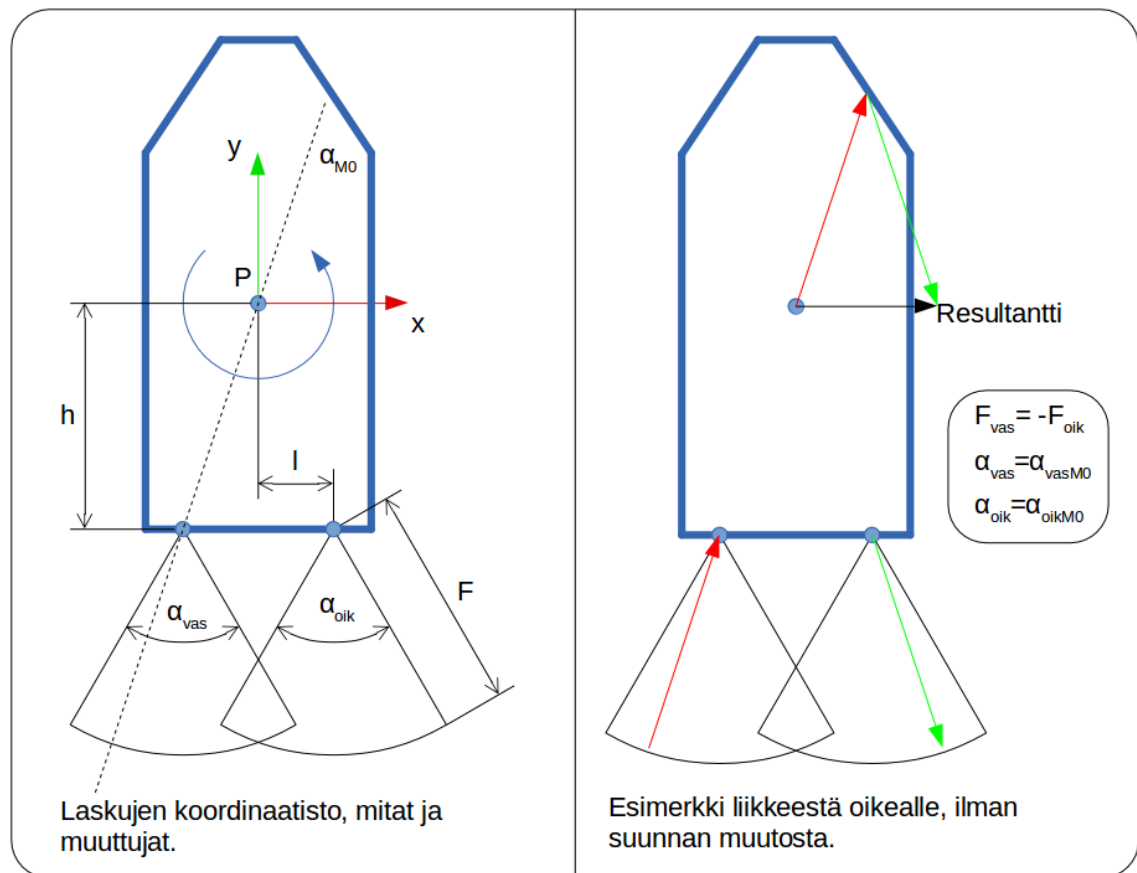
Kaaviossa on havainnoillistettu, kuinka järjestelmän ajotapa vaihtuu painonapista. Napin painallus on aina yksi painallus eli painalluksen pituudella ei ole merkitystä. Ajotapa vaihtuu nykyisestä ajotavasta aina nuolen osoittamaan ajotapaan. Vasemmalla puolella on punaisen painonapin siirtymiset ja oikealla puolella vihreän painonapin siirtymiset

3.1 Valmiustila

Valmiustilaan mennään telakointi-ajotavan kautta, tai jos ohjelmansisäinen vahtikoira-rutiini aktivoituu häiriötilanteen johdosta. Valmiustilassa moottorit eivät reagoi ohjausliikkeisiin tai muutoksiin moottoreiden asennossa. Tässä tilassa ei myöskään lueta ohjaussauvan asentoja, vaan ainoastaan tarkkaillaan painonappeja ja odotetaan uuden ajotavan valitsemista. Vahtikoira-rutiini nollautuu valmiustilassa käynnin yhteydessä, joten jos häiriö oli tilapäinen, ei järjestelmää tarvitse käynnistää kokonaan uudelleen.

3.2 Telakointi-ajotapa

Telakointi-ajotavan ollessa kytkettynä vene pyrkii vastaamaan ohjaussauvan liikkeitä. Tämä mahdollistaa käyttäjäystävällisen kokemuksen venettä telakoidessa. Ohjauksen määrittely nojautuu vektorien yhteenlaskuun ja voiman momentin määrittelemiseen. Näihin määritelmiin tulee tietää seuraavat veneen karakteristiset mitat: moottorin leveys- ja pituussuuntainen etäisyys veneen massakeskipisteestä (kuva 2). Näiden mittojen tulisi olla käyttäjän määriteltävissä siinä tapauksessa, että järjestelmää halutaan valmistaa erillisenä tuotteena.



Kuva 2: Laskut perustuvat tällaiseen järjestelmään (ei oikeat mittasuhteet)

Veneen x- ja y-suuntaiset komponentit määritellään seuraavilla voiman perusyhtälöillä.

$$F_x = \cos(\alpha_{vasen} + 90^\circ) \cdot F_{vasen} + \cos(\alpha_{oikea} + 90^\circ) \cdot F_{oikea}$$

ja

$$F_y = \sin(\alpha_{vasen} + 90^\circ) \cdot F_{vasen} + \sin(\alpha_{oikea} + 90^\circ) \cdot F_{oikea}$$

Termillä $+90^\circ$ kaavassa käännetään moottorin kulma veneen koordinaatistoa vastaavaksi. Kulma on asteina ymmärrettävyyden parantamiseksi. Konekielellä laskut suoritetaan radiaaneina.

$$M = \cos(\alpha_{vas} + 90^\circ) \cdot F_{vas} \cdot h - \sin(\alpha_{vas} + 90^\circ) \cdot F_{vas} \cdot l + \cos(\alpha_{oik} + 90^\circ) \cdot F_{oik} \cdot h + \sin(\alpha_{oik} + 90^\circ) \cdot F_{oik} \cdot l$$

Laskuissa positiivinen momentti on vastapäivään, positiivinen x-akseli oikealle ja positiivinen y-akseli ylös (kuva 2), kulmankäännön jälkeen venettä ylhäältäpäin katsottuna.

Kaavoista voidaan muodostaa yhtälöryhmä, joka kertoo veneeseen vaikuttavat voimat. Tätä yhtälöryhmää käytetään moottoreiden kulman ja tehon määrittämiseen.

$$\begin{cases} F_x = \cos(\alpha_{vasen} + 90^\circ) \cdot F_{vasen} + \cos(\alpha_{oikea} + 90^\circ) \cdot F_{oikea} \\ F_y = \sin(\alpha_{vasen} + 90^\circ) \cdot F_{vasen} + \sin(\alpha_{oikea} + 90^\circ) \cdot F_{oikea} \\ M = \cos(\alpha_{vas} + 90^\circ) \cdot F_{vas} \cdot h - \sin(\alpha_{vas} + 90^\circ) \cdot F_{vas} \cdot l + \cos(\alpha_{oik} + 90^\circ) \cdot F_{oik} \cdot h + \sin(\alpha_{oik} + 90^\circ) \cdot F_{oik} \cdot l \end{cases}$$

Muodostuneella yhtälöryhmällä ei ole yksiselitteistä ratkaisua, sillä se sisältää enemmän tuntemattomia tekijöitä kuin yhtälöitä. Tuntemattomien tekijöiden vähentämiseksi on tehtävä loogisia valintoja ohjausliikkeistä riippuen. Momentin yhtälöstä piirretty kuvaaja (liite 1) momentin ja kulman suhteen kuvaa moottoreiden aiheuttamaa momenttia, sekä nollamomentin kulman. Voima kuvaajassa on kerroin, sillä se on yhtälössä lineaarinen tekijä, toisin kuin kulma. Kuvaajasta huomataan momentin olevan lähes lineaarinen pienillä kulmilla. Kyseinen kuvaaja pätee vain Black Cat Boatsin katamaraaniin, sillä sen määrittelemiseen on käytetty kyseisen veneen mittoja.

Sivuttaisliikkeelle on voimassa seuraavat ehdot.

$$\begin{aligned} F_y &= 0 \\ F_x &= x \\ M &= 0 \end{aligned}$$

Muuttuja x on proportionaalinen ohjaussauvan asentoon x -akselilla. Kuvaajasta (liite 1) nähdään kummankin moottorin nollamomentin kulman olevan vastakkaisilla puolilla. Silloin toisen moottorin pakittaessa, kun toinen kaasuttaa vastaavalla voimalla, saadaan voiman y -komponentti kumotuksi, momentin pysyessä nollana. Kuvaajasta huomataan myös, että kulmien samassa suhteessa kasvattaminen ja pienentäminen aiheuttaa suunnasta riippuen momentin, voiman suunnan komponenttien pysyessä samana. Tämä on hyödyllinen tieto, kun halutaan samanaikainen kiertoliike sivuttaisliikkeeseen yhdistettynä. Momentin nollakulma esiintyy silloin, kun moottorin kulman tangentista piirretty vektori kulkee veneen massakeskipisteen läpi (kuva 2).

$$M_0 = \arctan\left(\frac{l}{h}\right)$$

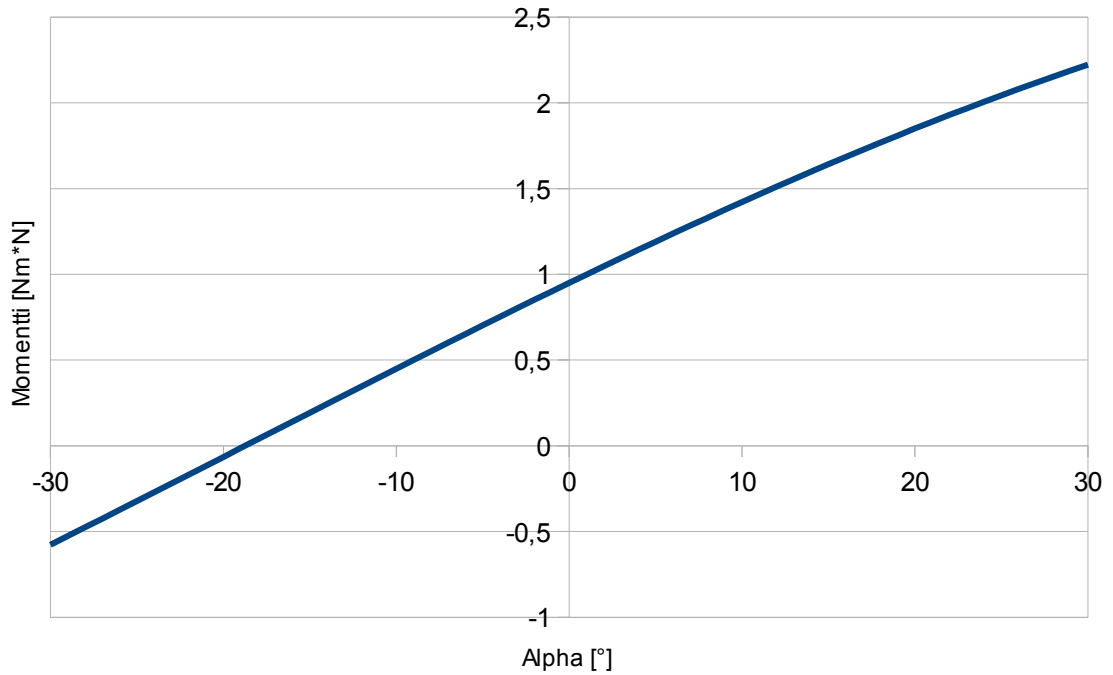
Tulee muistaa, että pituus l on suhteessa keskipisteeseen, joten toisella puolella venettä se esiintyy $-l$:nä (kuva 2).

Diagonaalinen liike noudattaa seuraavia ehtoja.

$$\begin{aligned} F_y &= F_x \\ M &= 0 \end{aligned}$$

Näillä ehdoilla voidaan aikaisemmin muodostetusta yhtälöryhmästä muodostaa yhtälöpari sekä ratkaista moottorien asennot ja suhteelliset voimat tietyillä loogisilla valinnoilla moottorien asentojen ja tehojen suhteen.

Momentin yhtälöön perustuvasta kaavasta on piirretty kuvaaja (kuva 3), johon sijoitettu lineaarinen regressio osoittaa virheen pienimmän neliön summaa soveltaen, että lineaarinen interpolaatio moottoreiden kulman muutosten välillä soveltuu järjestelmään 99 %:n tarkkuudella.



Kuva 3: Lineaarinen regressio sovitettuna momentin kuvaajaan

Tämä tarkkuus on riittävä, sillä suuremmat virheet ilmenevät asemointi-, laskenta- ja mittausvirheissä. Kyseessä on vene, joten täydellisen systeemin rakentaminen on hieman turhaa, johtuen suuresta hitaudesta ohjausliikkeissä.

Telakointi-ajotavassa on veneen kiertoliike suunnanohjauksen yhteydessä tai erikseen toteutettu seuraavasti. Kun tarvitaan pelkkä kiertoliike, veneeseen kohdistetaan momentti moottoreiden ollessa keskiasennossa toisen pakittaessa ja toisen ollessa kaasulla. Yhdistetty kiertoliike ja suunnanohjaus tapahtuu kuvaajan (liite 1) mukaisesti, joko lisäämällä moottoreiden kulman poikkeamaa keskiasennosta tai vähentämällä sitä, riippuen kiertoliikkeen suunnasta.

3.3 Ensimmäinen ajotapa

Ensimmäisessä ajotavassa moottoreiden teho vastaa ohjaussauvan y-akselin asentoa. Tehon säätö on dynaaminen, samalla tavalla kuin perinteisessä kaasuvivulla toimivassa veneessä. Erona perinteiseen veneeseen tässä on kuitenkin se, että ohjaussauvan asema palautuu keskiasentoon, jos sitä ei pidätellä asennossaan. Koska tämän tyyppisellä ohjausmekaniikalla ei saada samanlaista ohjaustuntumaa kuin

perinteisessä veneessä, on tähän järjestelmään ohjausmukavuuden lisäämiseksi kaksi erilaista ohjaustapaa, jotka eroavat toisistaan tehonsäädön perusteella. Tässä ajotavassa tehonsäätö on dynaaminen ja toisen ajotavan tehonsäätö on kiinteästi säädettävä.

Suuntaohjaus tapahtuu ohjaussauvan sivusuuntaisella liikkeellä (x-akselilla). Ilmiötä, jossa toinen moottori kulkee käännöksessä suuremman matkan kuin toinen, ei huomioida, sillä venettä eivät käännä niinkään moottorit, vaan veneen runko ja siihen aiheutettava momentti. Autoissa on tätä ilmiötä varten tasauspyörästä. Veneen kääntäminen tapahtuu kääntämällä molemmat moottorit suunnasta riippuen joko oikealle tai vasemmalle. Moottoreissa ei käännön aikana tapahdu tehon muutosta, sillä tämä aiheuttaisi tarpeetonta kuormaa moottoreille, koska ne joutuisivat jarruttamaan venettä epäsynkronisesti. Tästä moottorille aiheutuvaa kuormaa kasvattaa huomattavasti veneeseen sitoutunut liike-energia. Sen sijaan jarruttamiseen käytetään veneen omaa runkoa.

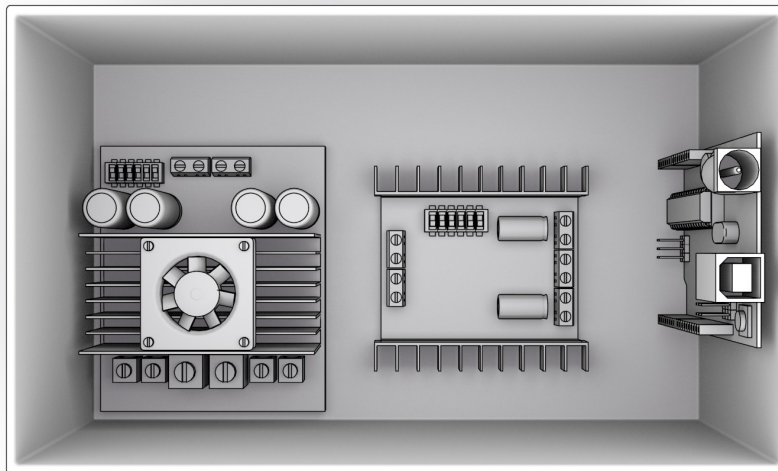
3.4 Toinen ajotapa

Toinen ajotapa muistuttaa perinteisen veneen ohjausta, siinä veneen nopeutta säädetään viemällä ohjaussauvan asento tietyn rajan yli y-akselilla. Ohjaussauvan asennon ollessa yli tehonmuutoksen rajan, muuttuvat moottoreiden tehot kiinteään prosentuaalisen osuuden verran moottoreiden maksimitehoista. Tehoa muutetaan tietyllä intervallilla niin kauan, kun ohjaussauvan asento ylittää muutoksen rajan. Näin saadaan helposti säädettyä haluttu nopeus. Tehon muutoksen suunta riippuu ohjausliikkeen suunnasta, eli tehoa voidaan joko kasvattaa tai pienentää. Suunnan pystyy myös vaihtamaan. Ohjaussauva palaa jousen avulla keskiasentoon, joten tehon säätö ei ole aivan yhtä intuitiivista kuin se on kaasuvivulla. Tilanteissa, joissa nopeus täytyy muuttaa nopeasti, on vaihdettava ajotapaa. Ajotavan voi vaihtaa vauhdissa ja se vaihtuu painonapista riippuen, joko ensimmäiseen ajotapaan tai telakointi-ajotapaan. Suuntaohjaus on sama kuin ensimmäisessä ajotavassa.

4 Toimilaitteet ja kokoonpano

Prototyypissä on 2 kpl Minn Kota Maxxum 70-perämoottoreita, joiden yhteenlaskettu työntö on 140 lbs eli noin 620 N. Kääntömoottoreina toimii Transmotecin (WRD63118)-moottorit, joissa on matoruuviperiaatteella toimiva alennusvaihte. Asennon tunnistukseen käytetään absoluutti-enkooderia, jossa on PWM-lähtö asennolle. Moottoreita käyttää Sabertoothin 2x12 ja 2x60-moottorinohjaimet (kuva 4). Jälkimmäinen luku ilmaisee syötettävän virran maksimiarvon lähtöä kohden. Vahvistimessa on kaksi moottorilähtöä. Pienempää moottorinohjainta käytetään kääntömoottoreille ja suurempaa perämoottoreille. Ohjelman suorittamisesta vastaa Arduino Uno R3-mikrokontrolleri (kuvat 4 ja 7). Ohjaaminen tapahtuu kolmiakselisella sauvaohjaimella ja ohjaustavan valintaan on kaksi mikrokytkintä.

BLACK CAT BOATS



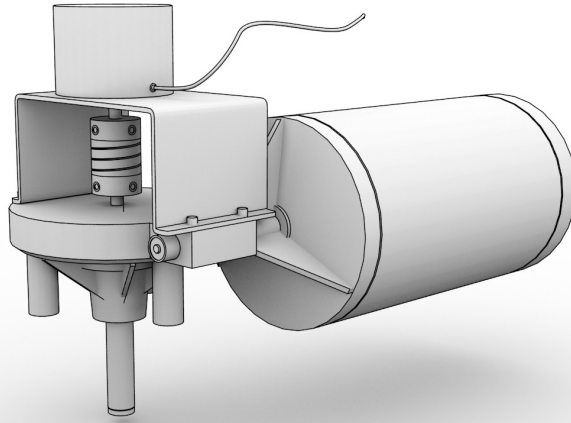
(havainnekuva)

Kuva 4: Komponentit vasemmalta: Sabertooth 2x60, Sabertooth 2x12 ja Arduino Uno

Suurin osa projektin kokoonpanosta on tehty aikaisempina TKI-projekteina ja Black Cat Boatsin henkilökunnan toimesta. Tässä työssä on korvattu induktiiviset anturit (6 kpl) kahdella absoluutti-enkooderilla. Muutoksen perusteena ovat seuraavat asiat: absoluutti-enkooderi vaatii vähemmän I/O-paikkoja kuin induktiiviset anturit, se toimii 3-5 V:n jännitteellä, joten sen jännitettä ei tarvitse alentaa, toisin kuin 24 V:n induktiivisten antureiden, absoluutti-enkooderi on helpompi asentaa ja ottaa käyttöön sekä on edullisempi kuin induktiiviset anturit. Absoluutti-enkooderit kiinnittyvät prototyypin

kokoonpanossa alennusvaihteen matoruuvin jälkeiseen toisiohammaspyörään (kuva 5), joka on suoraan kiinni moottorin akselissa. Tässä tulee huomata, että prototyypin kokoonpano ei välttämättä ole lopullinen kokoonpano.

BLACK CAT BOATS

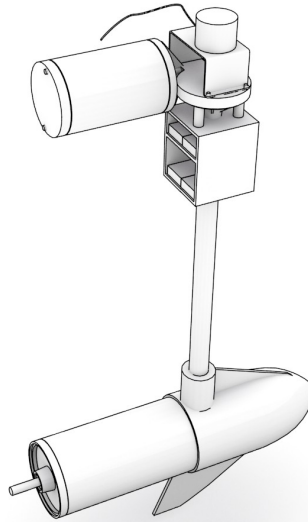


MOOTTORIN SUUNTAOHJAUS

Kuva 5: Transmotec-kääntömoottori matoruuvi alennusvaihteella, jonka päälle on kiinnitetty absoluutti-enkooderi (havainnekuva)

Projektia varten suunniteltiin kokonaan uudenlainen järjestelmä moottorin suunnanohjaukselle. Tämä järjestelmä kehiteltiin hihnavetoiseksi. Siinä absoluutti-enkooderin asema olisi ollut pienemmässä suhteessa, kuin moottorin todellinen asema. Tämä seikka on huomioitu ohjelmassa, jos myöhemmin vaihdetaan kokoonpanoa. PWM-tyyppisillä enkoodereilla ei kuitenkaan tule mitata asentoa, joka on enemmän kuin yhden kierroksen, sillä pulssin leveyden muuttuminen alkaa aina alusta samassa kohdassa. Uutta suunnitelmaa ei lähdetty toteuttamaan, sillä projektin tarkoituksena on kehittää ja testata moottorinohjausjärjestelmää. Testaaminen onnistuu nykyisen prototyypin kokoonpanolla (kuva 6).

BLACK CAT BOATS



(havainnekuva)

Kuva 6: Moottorikokoonpano. Kuvassa on tarvittavat moottorit ja anturit moottorinohjaukselle. Näitä tulee järjestelmään 2 kpl.

Sabertooth-moottorinohjaimet käsittelevät monenlaisia signaaleja analogisesta signaalista sarjaliikenteeseen asti. Ne käsittelevät myös R/C-ohjausta, joka tapahtuu PWM-signaaleilla. Moottoreita ohjataan pulssin leveyttä säätämällä. 1 ms:n pulssin päällä olo vastaa täyttä pakkia, 2 ms vastaa täyttä kaasua ja 1,5 ms:ssa moottorit ovat pysähtyneenä. Sabertooth-moottorinohjaimet konfiguroidaan dippikytkimien avulla ja siihen löytyy Sabertoothin valmistajan Dimension Engineeringin kotisivuilta ohjeet [1]. Moottorinohjaimet noudattavat mikrokontrolleriohjauksessa viimeisintä saatua käskyä ja tämän takia ohjelmassa on ns. vahtikoira-rutiini. Tämä pitää huolen siitä, että jos absoluutti-enkooderin johto esimerkiksi irtaantuu, niin vahtikoira pysäyttää kaikki moottorit, kunnes häiriö korjataan. Vahtikoira-rutiini on tässä yhteydessä ohjelmaan implementoitu metodi, eikä sitä tule sekoittaa prosessorin sisäiseen Watch Dog Timeriin (WTD).

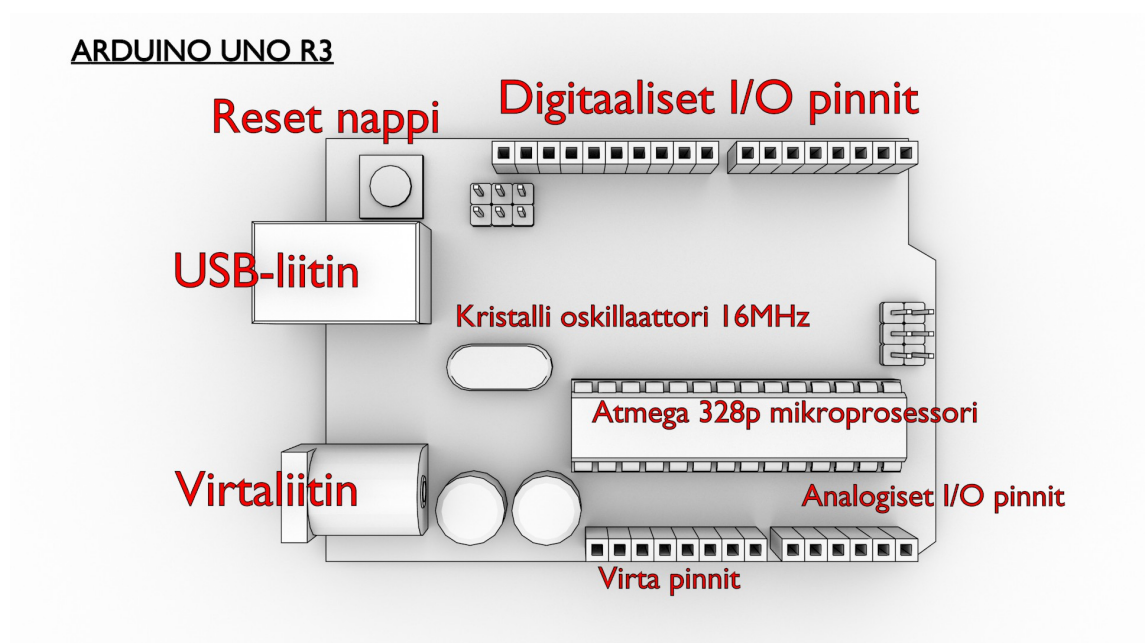
5 Ohjelma sulautetussa ympäristössä

Sulautetussa järjestelmässä tietokone ohjaa jotakin prosessia tai toimintaa. Tässä projektissa tietokoneeksi on valittu Arduino Uno R3-mikrokontrolleri. Projektin edetessä tulee rakentaa ohjelma mikrokontrollerille. Ohjelma rakennetaan yleensä

ohjelmointikielellä, joka tavanomaisesti on C++ tai jokin sen johdannainen. Yleisimmät ohjelmointikielet eivät juuri poikkea toisistaan. On myös olemassa erilaisia graafisia ohjelmointikieliä, kuten tikapuukaavio- ja funktioblokkiohjelmointi. Prototyyppialustoille kuten Arduino Uno on ominaista, että ne sisältävät ohjelmoijan (hardware, joka on integroitu alustalle) itsessään, joten erillistä ohjelmointilaitetta mikroprosessorin ohjelmointiin ei tarvita. Prototyyppialustojen ohjelmointi tapahtuu joko suoraan IDE:n välityksellä tai ns. drag and drop -tyyppisellä tavalla, jossa kirjoitettu ohjelma käännetään compilerissa konekielelle ja ulos tullut tiedosto vedetään tietokoneella porttiin, johon kehitysalusta on kiinnitetty.

5.1 Arduino Uno-mikrokontrolleri projektissa

Alustana projektissa toimii Arduino Uno R3, jossa on Atmega328p-prosessori. Prosessori on 8-bittinen ja kellotaajuus on 16 MHz [2]. Suurin mahdollinen kellotaajuus ko. prosessorille on 20 MHz [3, s. 1]. Lähtökohtana prosessorin vaatimukselle suorituskyvyn suhteen on ollut yksi ohjelmankierto kahdessa millisekunnissa. Ohjelman vasteajaksi on valittu 2 ms, sillä se on sama kuin nykyisissä tietokoneen näytöissä ja sitä ei käytännössä huomaa. Karkeasti Arduino kykenee laskemaan 2 560 000 bittiä kyseisessä aikarajassa (kuva 7).



Kuva 7: Arduino Uno R3. Kuvassa esitetty tärkeimmät peruspiirteet mikrokontrollerista.

$$\frac{8 \cdot 16\,000\,000}{\frac{1}{0.02}} \rightarrow 8 \cdot 16\,000\,000 \cdot 0.02 = 2\,560\,000$$

5.2 Arduino IDE

Yleisesti ottaen Arduinon IDE:tä on pidetty hitaana tiedon I/O-pinneiltä kirjoittamisen ja lukemisen suhteen. Hitaus ei synny kehitysympäristössä, vaan yleensä tarpeettoman monimutkaisesta tavasta, jolla compileri kääntää ohjelman konekielelle. Tässä projektissa ei ilmennyt hitauden suhteen ongelmaa, joka olisi vaatinut suuremman lähestymisen I/O-kohteisiin. Ohjelman kiertoon kuluva aika on rutiinista riippuen n. 300 - 1400 µs eli hitaimmillaan 1,4 ms, mikä on alle vaaditun 2 ms:n. PWM-signaalin lukemiseen absoluutti-enkooderilta ilmeni hitautta, sillä tavanomaisesti se vaihtelee 1 - 2 ms:n pituudella ja tämän ajan prosessori odottaa tekemättä mitään, ennen kuin se on mitannut pulssinleveyden. Kahdelle absoluutti-enkooderille tämä aiheuttaisi siis huonossa tapauksessa 4 ms:n lisäviiveen. Absoluutti-enkooderien lukeminen on tämän takia siirretty rinnakkaiseksi toiminnoksi, joka keskeyttää pääohjelman aina, kun se saa tuloksen pulssinleveydestä tai aloittaa uuden pulssin mittaamisen [4]. Tämä toiminto ei aiheuta lisäviivettä ohjelman kiertoon. Keskeyttäminen suoritetaan Interrupt-rutiinilla, joita Arduino Unossa on kahdessa pinnissä (oletuksena 2 ja 3, mutta voidaan siirtää).

Arduino kehitysympäristönä on laaja ja hyvin dokumentoitu [5]. Sen syntaksi on C/C++. Mikäli myöhemmin ilmenee tarvetta, on lähdekoodi helposti käännettävissä muihin kehitysympäristöihin toisille alustoille esim. ARM. Tällainen tarve voi ilmetä, jos halutaan siirtyä ns. älykkäämpiin järjestelmiin, jotka sisältävät useampia mikrokontrollereita tai halutaan sulauttaa ohjelma muihin veneen järjestelmiin. Useimmat ARM-prosessoriarkkitehtuurit tukevat yleisesti älykkäissä järjestelmissä suosittua CAN-väylää, mitä Arduino ei tue.

Yleisesti hyvänä käytäntönä tällaisessa ohjelmoinnissa, jossa ei rakenneta ohjelmaa kerralla alusta loppuun, se kannattaa tallentaa uudella nimellä (kopioda vanha) jokaisella kerralla, kun jatkaa ohjelmaa. Tällainen käytäntö on hyvä siltä varalta, jos tiedosto korruptoituu (kävi kerran projektin aikana) tai tekee epähuomiossa jotain vahinkoa ohjelmalle, koska silloin ei tarvitse aloittaa kokonaan alusta. Kannattaa myös välillä tehdä varmuuskopioita toiselle kiintolevylle tai muistitikulle mahdollisen järjestelmävirian varalle.

Arduino kykenee laskemaan kaikki tavanomaiset matemaattiset operaatiot, polynomiregressiot, trigonometriset funktiot jne. Näitä laskettaessa on kiinnitettävä huomiota kahteen seikkaan. Ensimmäiseksi muuttujan, jolle lasku suoritetaan, on oltava oikeaa tyyppiä ja sen rajojen on oltava tuloksen sisällä. Esim. int- tyyppin muuttujan rajat ovat -32 768- 32 768, kun taas etumerkitön unsigned int on välillä 0- 65 535. Toiseksi tapauksessa, jossa käytetään jotain Arduino IDE:n sisäistä funktiota luvun käsittelyyn, tulee sen referenssistä tarkastaa, minkälaisia arvoja se kykenee käsittelemään. Suurin osa laskuvirheistä johtuu vääränlaisesta muuttujasta.

Arduino ei kykene ratkaisemaan lineaarisia tai epälineaarisia yhtälöryhmiä. Tähän keinona on selvittää yhtälöryhmän tulos tietyillä tiedetyillä arvoilla ja määritellä interpolaatio näiden arvojen välillä. Interpolaatio voi olla lineaarinen tai polynominen. Yhtälöryhmän laskeminen on mahdollista Arduinolla käyttäen sisäkkäisiä silmukoita, joissa lasketaan tulos jokaiselle arvolle valitulla inkrementillä. Tällaisessa laskussa tulee yhtälöryhmän yhtälöiden lopputuloksen olla nolla ja Arduino valitsee arvot, joiden itseisarvojen summa on mahdollisimman lähellä nollaa. On hyvä muistaa, että jokainen yhtälö yhtälöryhmässä lisää laskentakuormaa eksponentiaalisesti, samoin kuin inkrementin pienentäminen desimaalilla. Tällaista laskutoimitusta ei Arduinolla kykene suorittamaan ohjelman ajon aikana, sillä se saattaa viedä aikaa parista sekunnista useampaan minuuttiin (tässä järjestelmässä alustus kestää hieman reilu 2 s). Tämä tulee siis suorittaa alustuksessa, kuten tässä projektissa on tehty tai tapauksessa, jossa tulos on aina sama, kannattaa lasku suorittaa etukäteen muualla.

6 Jatkokehitysmahdollisuudet

Tämän työn ulkopuolelle jäi joitakin asioita, joita tulee kehittää. Tällaisessa järjestelmässä on paljon kehitysmahdollisuuksia ja ennen kuin kehitystyötä aloittaa, tulee tarkkaan suunnitella, minkälainen järjestelmä halutaan saavuttaa. Tässä kappaleessa listataan joitakin noista mahdollisuuksista. Listatuista mahdollisuuksista joitakin yritettiin toteuttaa, mutta ne jäivät toteuttamatta ajanpuutteen vuoksi. Osaa niistä myös suunniteltiin, mutta ne jätettiin toteuttamatta tässä vaiheessa tarpeettomina. Jatkokehitys on yhdeltä henkilöltä hidasta, joten kehityskohteet tulee ajatella modulaarisesti, eli yksi henkilö jokaiselle kehityskohteelle, ja myöhemmin ne yhdistetään yhdeksi kokonaisuudeksi. Ryhmätyön onnistumisen kannalta ryhmän jokaisen henkilön tulee tuntea nykyisen lähdekoodin perusperiaate ja oma

kehityskohteensa. Ryhmässä tulee myös olla johtaja, joka pitää huolen aikatauluista ja siitä, että moduulit pysyvät yhteensopivina. Ryhmänjohtaja kokoaa yhteistyössä muiden jäsenten kanssa moduulit yhdeksi kokonaisuudeksi ja huolehtii riittävästä informaation kulusta.

6.1 Näyttö ja GUI

Ohjauspaneeliin kannattaa sijoittaa näyttö, josta voi lukea moottoreiden asentoa, tehoa ja akkujen varaustasoa sekä muita hyödyllisiä tietoja. Pääasiallinen hyöty tällaisesta näytöstä kuitenkin olisi se, että sillä voisi säätää erilaisia muuttujia, kuten trimmata moottoreiden asennot. Näytön avulla voisi myös asettaa veneen karakteristiset ominaisuudet siinä tapauksessa, että järjestelmää käytetään erillisenä järjestelmänä toisissa veneissä, eikä vain Black Cat Boatsin katamaraanissa. Ominaisuuksien asettamiseen kannattaisi valita sellainen näyttö, joka sisältää myös kosketusominaisuuden, sillä ne eivät hinnaltaan juuri eroa tavallisesta näytöstä. Lähdekoodiin tällainen lisäys vaatisi GUI:n ohjelmoinnin ja trimmattavaksi valittujen muuttujien sijoittamisen mikroprosessorin pysyvään EEPROM-muistiin, josta ne voidaan lukea ajonaikaiseen RAM-muistiin alustuksen yhteydessä.

6.2 Taivasankkuri

Taivasankkuri voisi olla mahdollisuus tehokkaammilla moottoreilla. Taivasankkurin implementointi vaatisi GPS-paikantimen lisäyksen järjestelmään. Sijaintitiedot GPS-moduulista tulevat NMEA-lauseina, joista voidaan lukea monta erilaista parametria, kuten sijainti, aika, nopeus, suunta sekä paljon muuta. Tässä tapauksessa sijainti voidaan asettaa siihen hetkeen, kun taivasankkuri kytkettiin päälle ja uudet sijainnit ovat poikkeama, joka muunnetaan suuntaohjeiksi. Telakointi-ajotavan algoritmi veneen ohjaukseen sopii taivasankkurointiin tietyin muutoksin. Suuntaohjeet tulee suodattaa sopivalla tavalla kuten PID-säädöllä, sillä muutoin syntyy värähtelyä.

6.3 GPS ja modulaarinen ohjelma

Lähdekoodista voi myös kehittää modulaarisen, eli se haistelee, onko järjestelmään kytketty GPS-moduulia. Jos on, niin se mahdollistaa GPS:n käytön automaattisesti.

Pidemmälle kehitetyssä järjestelmässä voi myös käyttää karttoja reitin määrittelyyn tai tallentaa kuljetun reitin lokiin. Tällaista lokitiedostoa ei kuitenkaan kannata tallentaa prosessorin sisäiseen EEPROM-muistiin, sillä se ei kestä lukemattomia kertoja tallentamista [3, s. 1]. Tällaista lokitiedostoa varten tulee käyttää erillistä muistikorttia. Näyttöön voi lisätä ajan ja nopeuden sekä suuntiman tiedot GPS:n avulla.

6.4 Mikrokontrollerit ja lähdekoodin kääntäminen

Arduino Unossa on melko rajallinen määrä I/O-portteja. Jos järjestelmään kehitellään useita uusia ominaisuuksia, loppuvat nämä paikat melko nopeasti. Arduino tukee joitakin väylätekniikoita kuten I²C ja SPI. Tämä mahdollistaa useamman Arduinon kytkemisen samaan järjestelmään. Helpommaksi ratkaisuksi tulee kuitenkin ottaa käyttöön Arduino Mega, jossa on enemmän I/O-portteja. Toinen vaihtoehto on myös siirtyä kokonaan toisenlaiseen prosessoriarkkitehtuuriin, kuten ARM. Tässä tapauksessa lähdekoodin joutuu kääntämään, sillä käytettävät kirjastot ja funktiot eivät ole samoja uudella kielellä. Kääntäminen uudelle kielelle ei kuitenkaan ole välttämättä iso projekti, sillä lähes kaikki lähdekoodin funktiot ovat hyvin tavanomaisia ohjelmoinnissa ja ne löytyvät useammilla kielillä.

6.5 Järjestelmäkokoontamisen muutokset

Tämänhetkistä kokoonpanoa ei ole vielä optimoitu kustannusten ja toimivuuden suhteen. Moottoreiden suunnanohjaus kannattaa toteuttaa toisella tavalla. Hyvä tapa voisi olla hihnavetoinen järjestelmä, koska siinä voidaan helposti vaikuttaa välityssuhteeseen, jolloin ohjaustarkkuus paranee eikä tarvita niin tehokkaita moottoreita liikkeen aikaan saamiseen.

Kääntömoottoreiksi suljetun silmukan hybridi servomoottorit saattaisi olla helpoin vaihtoehto kokoonpanokustannusten kannalta. Tällaisia servomoottoreita valmistaa esimerkiksi Leadshine. Ne ovat edullisia ja niissä on asennontunnistus samassa. Yleensä tällaiset servot myydään yhdessä niihin kuuluvan moottorinohjaimen kanssa. Hybridi-servomoottorin asennontunnistus perustuu inkrementaali-enkooderiin absoluutti-enkooderin sijaan, joten järjestelmään täytyisi myös lisätä kotiaseman tunnustus. Tällaisen moottoripaketin käyttäminen tarkoittaisi, että enää ei tarvita erillistä

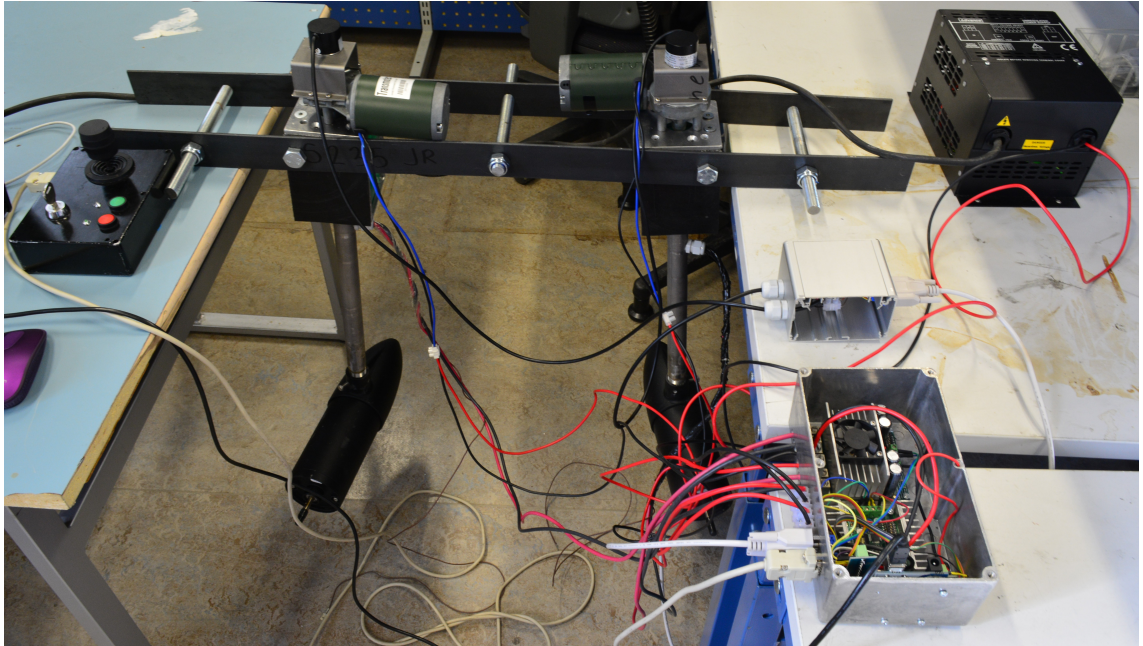
asennontunnistinta tai 2x12 Saberertooth-moottorinohjainta. Leadshinen moottorinohjain sisältää myös sisäänrakennetun vikatilan tunnistuksen, joten sen käyttö on turvallista. Vaihtaminen tällaiseen järjestelmään säästää rahaa ja vähentää kokoonpanoon kuluva-aikaa.

Moottoreiden kotelointi täytyy uusida tapauksessa, jossa järjestelmä vaihdetaan hihnavetoiseksi, sillä tällainen järjestelmä tarvitsee suuremman tilan ja hihna tulee saada suojaan vedeltä. Nykyinen kotelo on valmistettu hitsatuista alumiinilevyistä ja on työläs valmistaa. Kotelo kannattaa suunnitella käyttäen mahdollisimman paljon jo myytävänä olevia komponentteja ja helposti kokoonpantavaksi. Myös valukotelon suunnittelua kannattaa harkita. Akseli kannattaa myös laakeroida koteloon.

7 Prototyypin testaus

Työn edetessä rakennettiin erilaisia pienoistyyppityyppejä lähdekoodin testaamista varten. Näillä pienoistyyppityypeillä saatiin lähdekoodi konstruoidua ja testattua helposti.

Täyskokoisen prototyypin testauksessa (kuva 8) ilmeni, että kääntömoottorin ja akselin yhdistävässä laipassa on huomattavan suuri välys (n. 20°) molemmissa moottoreissa. Tämä välys on saatava pois, ennen kuin järjestelmää testataan käytännössä, sillä ohjausjärjestelmä ei asennoidu oikein sen takia.



Kuva 8: Täyskokoinen prototyyppi testipenkissä.

Järjestelmässä oleva Arduino oli myös osittain viallinen, joten se vaihdettiin toiseen samanlaiseen. Tämä vika aiheutti sen, ettei toinen moottori saanut tietoa asennostaan.

Lähdekoodi toimi odotetulla tavalla, sillä siitä oli pienoistyyppillä saatu kaikki ohjelmointivirheet poistettua. Lähdekoodiin tehtiin kuitenkin pieniä muutoksia ja trimmattiin kääntömoottoreiden proportionaalivahvistus sopivaksi tälle kokoonpanolle. Proportionaalivahvistukseen lisättiin myös kymmenesosa-asteen tyhjä alue, jolla vahvistimelle ei lähde ohjauskomentoja. Tämä säästää hieman virtaa.

Ohjelmaan lisättiin myös syklinen tehonmuutos, sillä Sabertoothin sisäänrakennettu (valinnainen) kiihdytysramppi ei ollut tarpeeksi loiva, mikä aiheutti sen, että moottorit vastasivat tehonmuutokseen liian nopeasti. Tämä aiheuttaa kuormaa moottoreille ja vahvistimelle. Sabertoothin rammit ovat konfiguroitavissa erillisellä ohjelmalla, mutta syklisen tehonmuutoksen johdosta sitä ei tarvitse tehdä.

Vahtikoira-rutiini testattiin irrottamalla absoluutti-enkoodereiden 9-pin d-liitin, ja se toimii myös oikein. Ohjelma sisältää lukuisia trimmattavia muuttujia käytön helpottamiseksi.

8 Yhteenveto

Insinööriyön tarkoituksena oli kehittää moottorinohjausjärjestelmää, ja se kyllä kehittyi siinä määrin, että aiemmin toimimattomasta järjestelmästä on tämän työn jälkeen päästy toimivaan järjestelmään. Järjestelmän sähkökaaviot (liite 2) uusittiin projektin lopuksi.

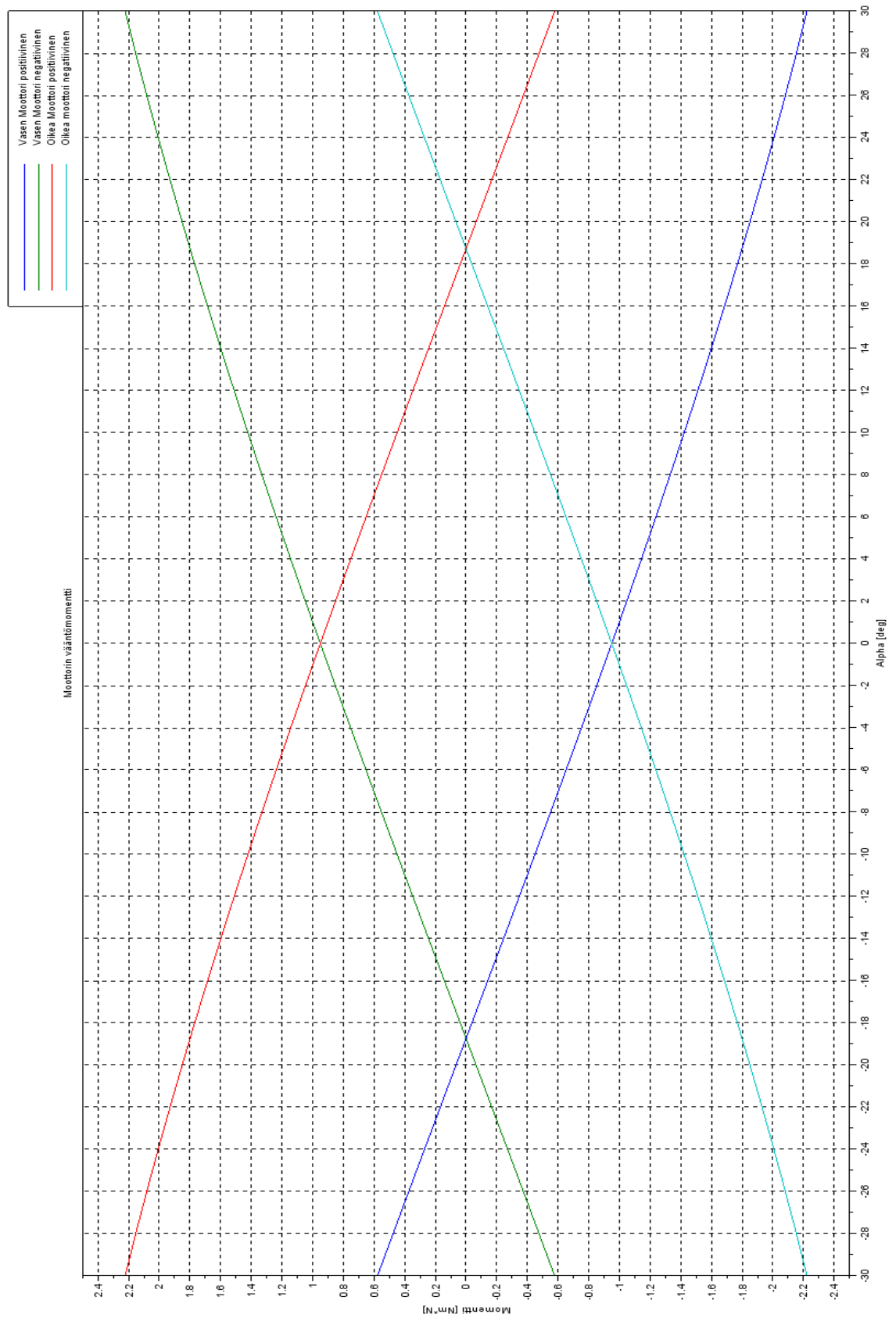
Insinööriyössä yritettiin saada näyttöjä toimimaan ja tehdä GUI, josta voisi konfiguroida tärkeitä muuttujia, sekä näytettyä erilaisia tietoja käyttäjälle järjestelmän tilasta. Tämä yritys jäi kuitenkin tältä osalta vajaaksi, koska tavarantoimittajien näytöt olivat yhteensopimattomia SPI-protokollaan tai toimimattomia saapuessaan.

Insinööriyössä testattiin myös GPS-moduulia mahdollisen taivasankkurin varalle, mutta ko. GPS-moduuli ei saanut asematietoa riittävän luotettavasti. Taivasankkurijärjestelmää ei lähdetty sen enempää tavoittelemaan, sillä epäiltiin, ettei moottorijärjestelmä ole riittävän tehokas sellaista varten. Jos GPS-moduulia olisi käytetty projektissa, olisi NMEA:ltä täytynyt ostaa standardi, joka määrittelee GPS-moduulin vastaanottamat lauseet [6].

Lähteet

- 1 Sabertooth DIP switch wizard. Verkkodokumentti. <http://www.dimensionengineering.com/datasheets/SabertoothDIPWizard/start.htm> Luettu: 21.4.2015.
- 2 Arduino Uno. Verkkodokumentti. <http://www.arduino.cc/en/main/arduinoBoardUno> Luettu: 21.4.2015.
- 3 Atmel ATmega datasheet. Verkkodokumentti. http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf Luettu: 21.4.2015.
- 4 Three Ways To Read A PWM Signal With Arduino. Verkkodokumentti. <http://www.benripley.com/diy/arduino/three-ways-to-read-a-pwm-signal-with-arduino/> Luettu: 21.4.2015.
- 5 Arduino reference. Verkkodokumentti. <http://www.arduino.cc/en/Reference/HomePage> Luettu: 21.4.2015.
- 6 NMEA standardit. Verkkodokumentti. <http://www.nmea.org/> Luettu: 21.4.2015.

Moottoreiden momentin jakautuminen Black Cat Boats katamaraanissa



Sähkökaavio

